

NS2 Project Report

Paper Name: CUBIC-FIT: A High Performance and TCP CUBIC Friendly Congestion Control Algorithm

https://www.researchgate.net/publication/260533734_CUBIC-FIT_A_high_performance_and_tcp_CUBIC_friendly_congestion_control_algorithm

Ayesha Binte Mostofa

Student ID : 1805062

Department of Computer Science and Engineering

February 28, 2023

1. Overview of the Proposed Algorithm Proposed Algorithm

CUBIC-FIT

This is a modified congestion control algorithm of TCP CUBIC (default congestion control algorithm in NS2). Cubic-fit is a delay-based TCP to extend the CUBIC algorithm framework. It simulates N no of flow in a single TCP connection to fully utilize network capacity.

Improvement:

- Performance over large range of network
- Throughput performance over wireless network
- Decrease end-to-end delay
- Maintain graceful friendliness with plain CUBIC networks

Assigned Network - Student ID: 1805062

$1805062 \bmod 8 = 6$

Network:

1. Wired
2. Wireless (802.11) (Static)

2. Modification made

2.1. In Equation:

In Equation:

In CUBIC:

$$\omega_{cubic} = C(t - I)^3 + \omega_{max} \quad (1)$$

$$I = \sqrt[3]{\omega_{max}b/C} \quad (2)$$

In CUBIC-FIT:

$$\omega_{cubic}^{fit} = 0.4(Nt - I)^3 + \omega_{max} \quad (3)$$

$$I = \sqrt[3]{10\omega_{max}/19N + 1} \quad (4)$$

where

$$\frac{C(4 - b)}{4b} = 1.9N^4 \quad (5)$$

$$N_{t+1} = \max\{1, N_t + 1 - \frac{RTT_t - RTT_{min}}{\alpha \cdot RTT_t} N_t\} \quad (6)$$

$$\alpha = \min\{\frac{1}{10}, \frac{RTT_{max} - RTT_{min}}{2RTT_{max}}\} \quad (7)$$

2.2. In Simulator:

Modification 1

```
    } else {  
        /* Compute new K based on  
        * (wmax-cwnd) * (srtt>>3 / HZ) / c * 2^(3*bictcp_HZ)  
        */  
        //cubic commented out  
        //ca->bic_K = cubic_root(cube_factor * (ca->last_max_cwnd - cwnd));  
        //-----  
        //cubicFit modification  
        ca->bic_K = cubic_root(10 * (ca->last_max_cwnd - cwnd) / (19*ca->flow_num_behavior + 1));  
        //-----  
  
        ca->bic_origin_point = ca->last_max_cwnd;  
    }
```

Modification 2

```
24 //-----  
25 //cubicFit modification  
26 if (t < ca->bic_K) /* t - K */  
27     offs = ca->bic_K - (ca->flow_num_behavior * t) ;  
28 else  
29     offs = (ca->flow_num_behavior * t) - ca->bic_K;  
30 //-----  
31  
32 /* c/rtt * (t-K)^3 */  
33 //delta = (cube_rtt_scale * offs * offs * offs) >> (10+3*BICTCP_HZ);  
34 //-----  
35 //cubicFit modification  
36 delta = (4/10 * offs * offs * offs) >> (10+3*BICTCP_HZ);  
37 //
```

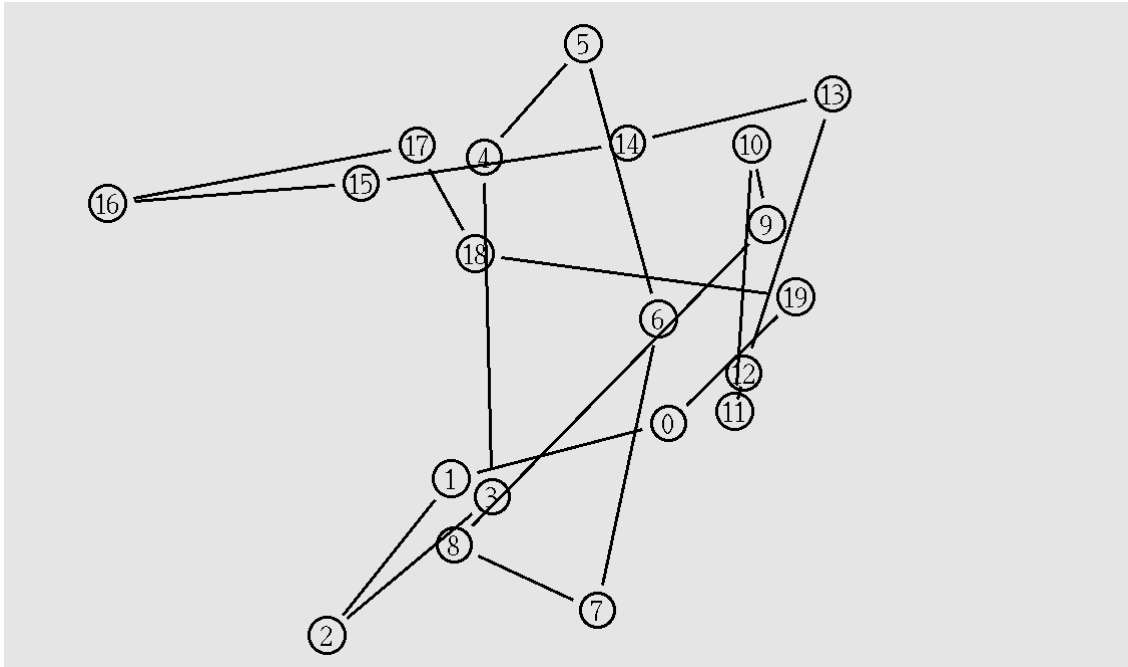
Modification 3

```
//cubicFit algo  
/* first time call or link delay increases */  
if (ca->delay_max == 0 || ca->delay_max < delay)  
    ca->delay_max = delay;  
  
//printf("%d delay_max\n", ca->delay_max);  
  
double alpha = .1;  
  
//printf("%lf alpha\n", alpha);  
  
double temp_alpha = .5 * (ca->delay_max - ca->delay_min)/ca->delay_max;  
  
//printf("%lf temp_alpha\n", temp_alpha);  
  
if(alpha > temp_alpha)  
    alpha = temp_alpha;  
  
//printf("%lf alpha\n", alpha);  
double flow = ca->flow_num_behavior;  
ca->flow_num_behavior = 1;  
//number of flows counting  
if(alpha != 0 && delay != 0)  
    flow = flow + 1 - flow*(1/alpha - ca->delay_min/(alpha * delay));  
if( ca->flow_num_behavior < flow)  
    ca->flow_num_behavior = flow;  
  
//printf("%d flow number\n", ca->flow_num_behavior);
```

3. Network topologies under simulation

3.1 Wired

3.1.1 Topology used: Mesh Topology



3.1.2 Configuration

- Simulation Time: 10 Seconds

3.1.3 Parameters Varied

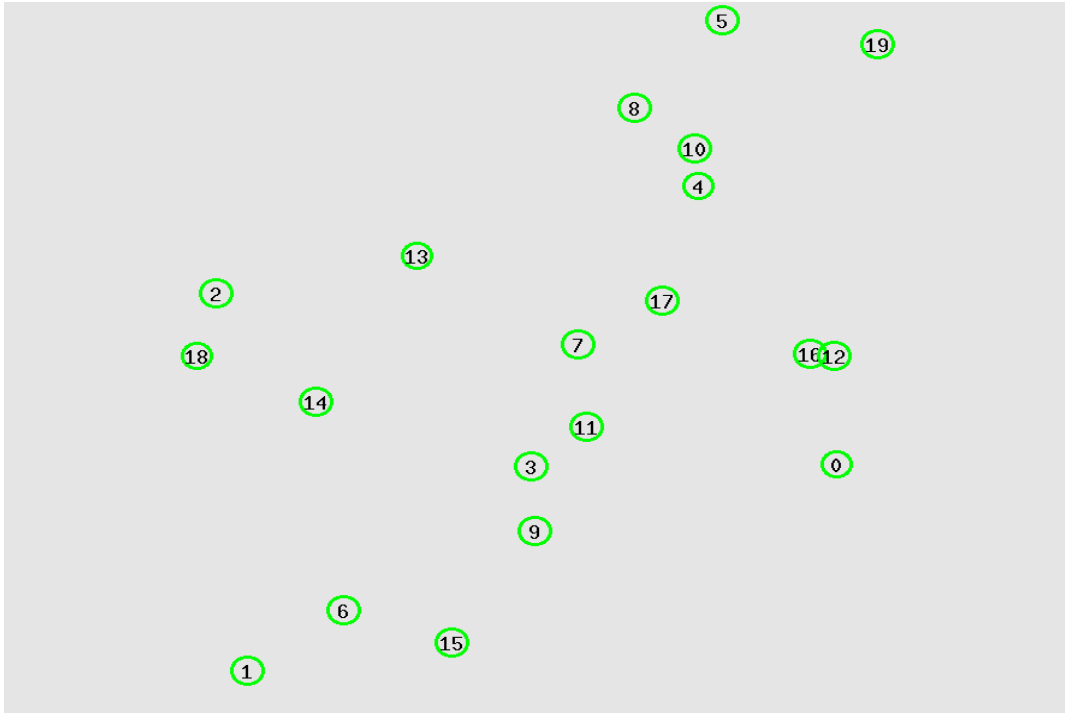
- Number of Nodes
- Number of Flows
- Number of Packets Per Second

3.1.4 Metrics Calculated

- Network Throughput
- End to End Delay
- Packet Delivery Ratio
- Packet Drop Ratio

3.2 Wireless (802.11) (Static)

3.2.1 Topology used: Random



3.2.2 Configuration

- Simulation Time: 10 Seconds

3.2.3 Parameters Varied

- Number of Nodes
- Number of Flows
- Number of Packets Per Second
- Coverage Area

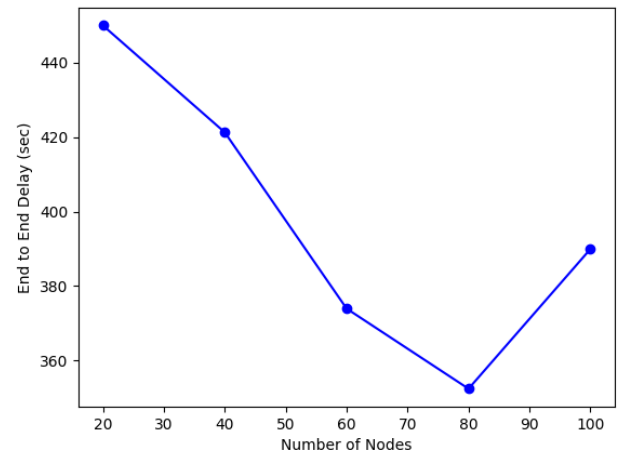
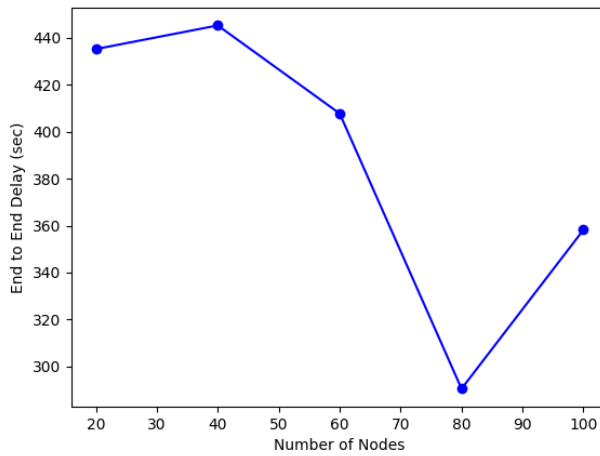
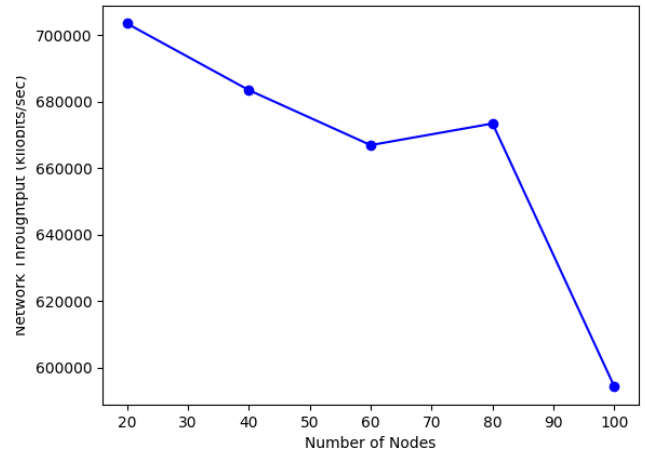
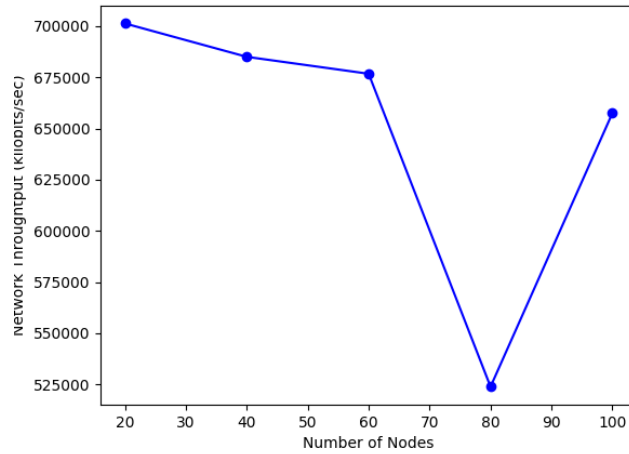
3.2.4 Metrics Calculated

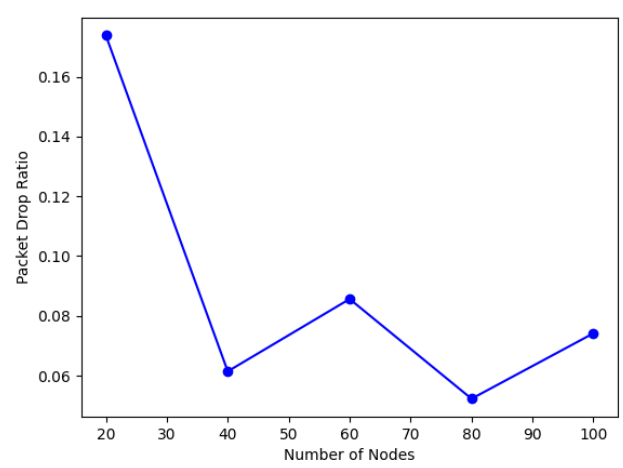
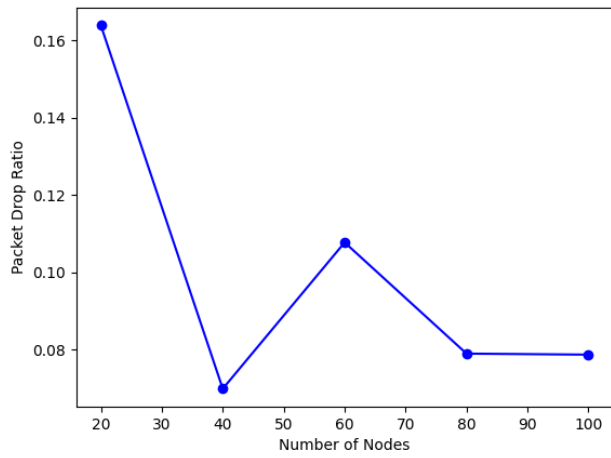
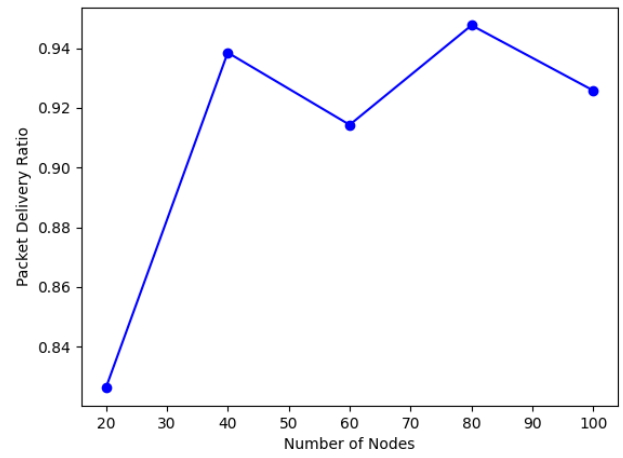
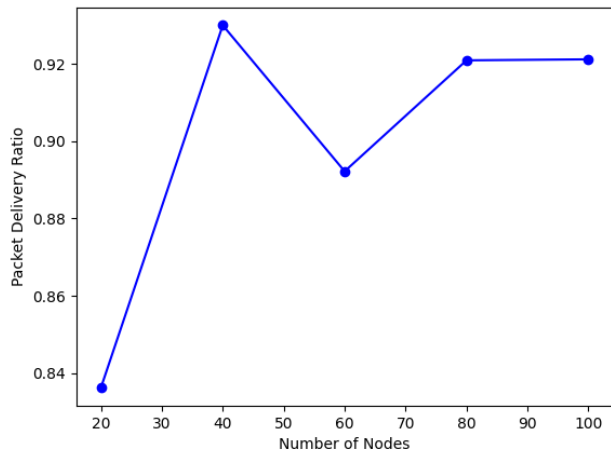
- Network Throughput
- End to End Delay
- Packet Delivery Ratio
- Packet Drop Ratio
- Total Energy Consumption

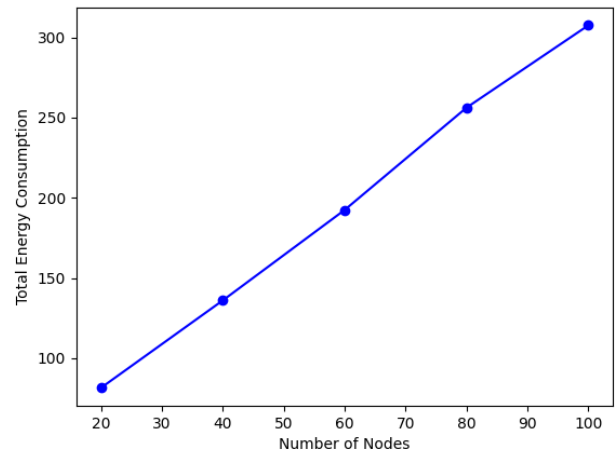
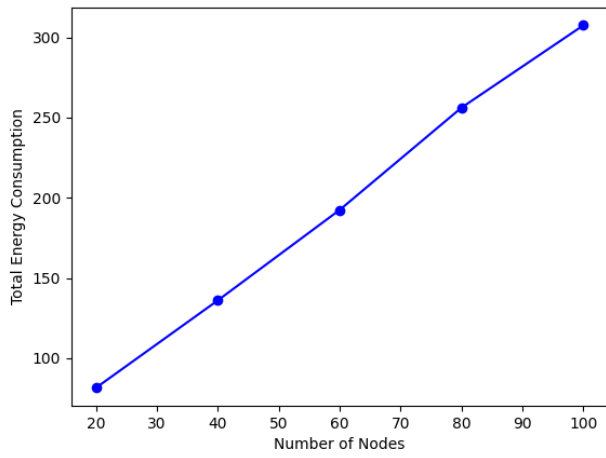
4. Graphs:

A. Wireless (prev vs modified)

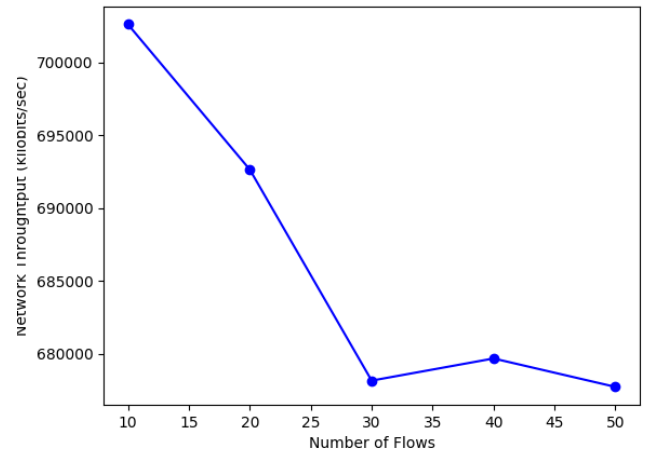
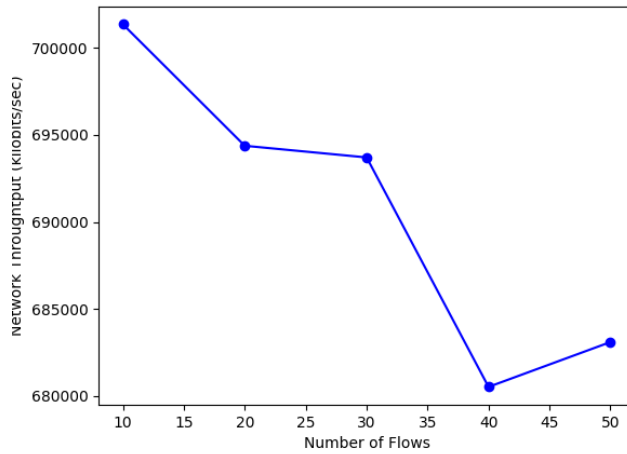
- Metrics Varying Number of Nodes:

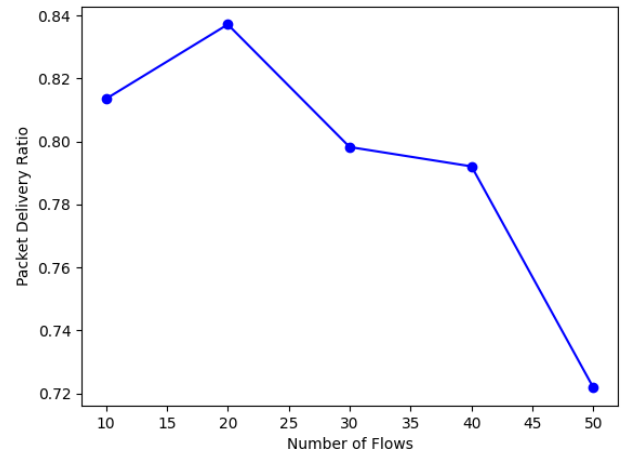
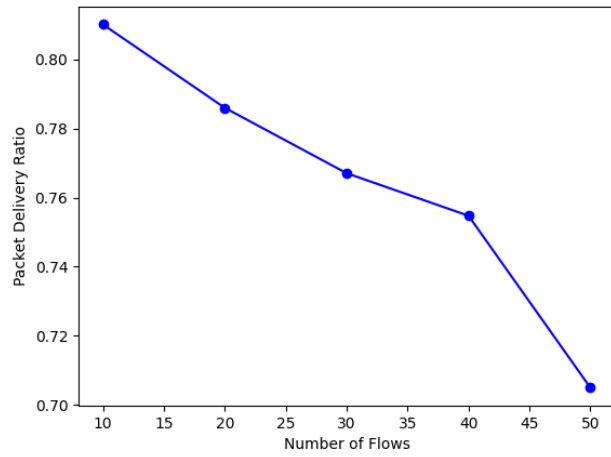
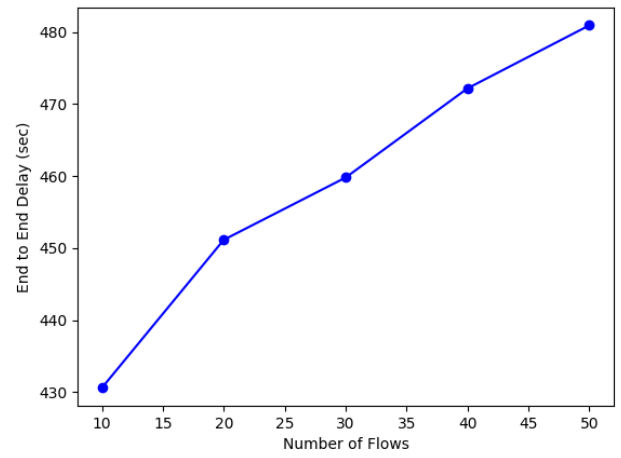
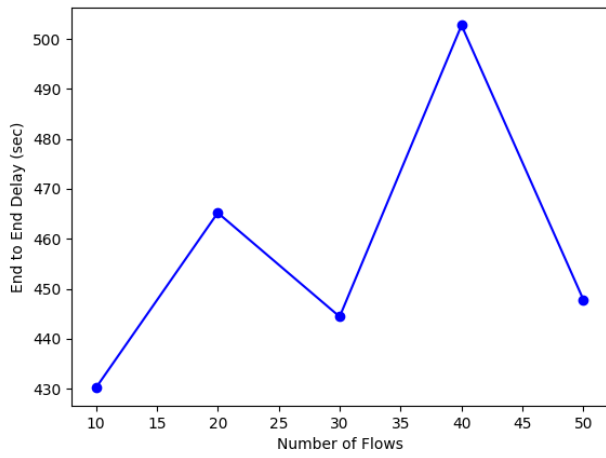


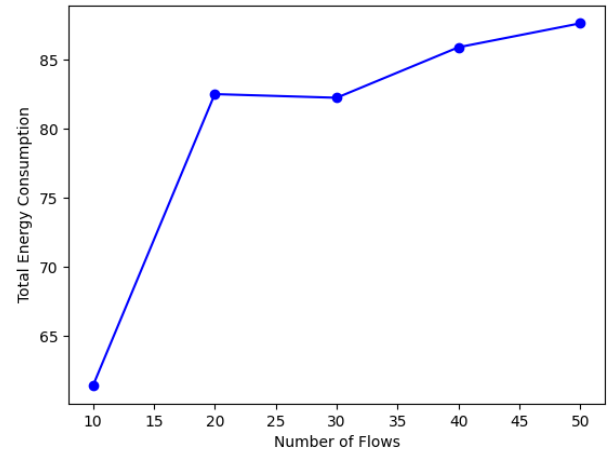
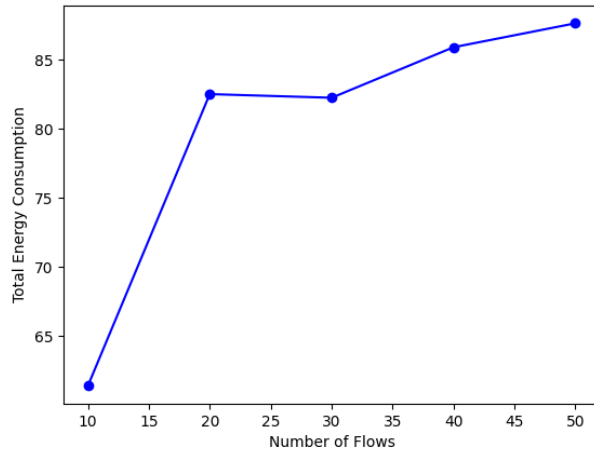
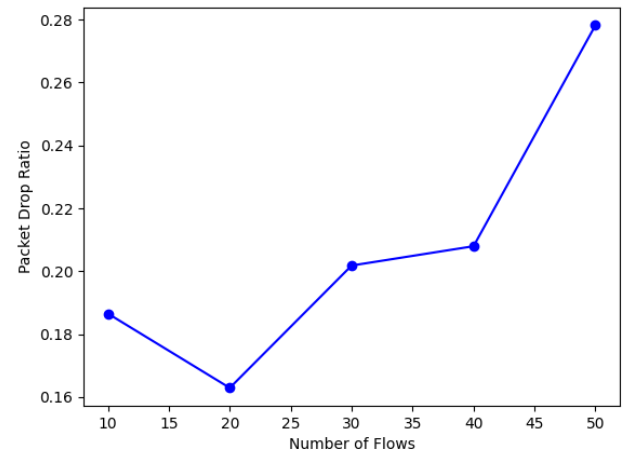
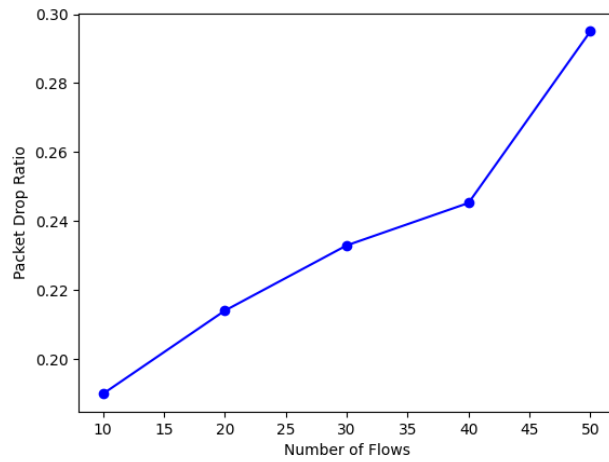




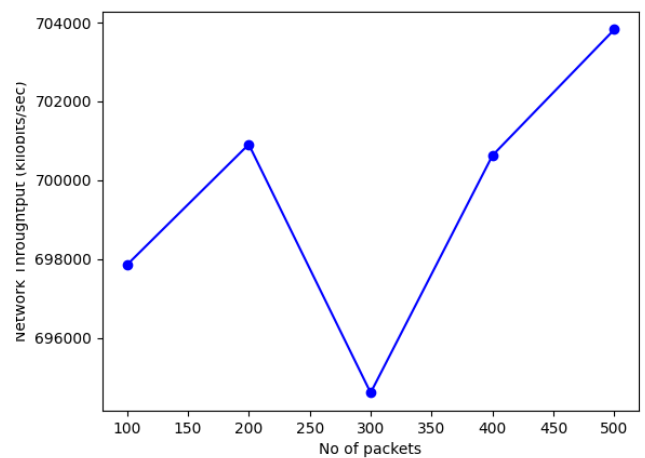
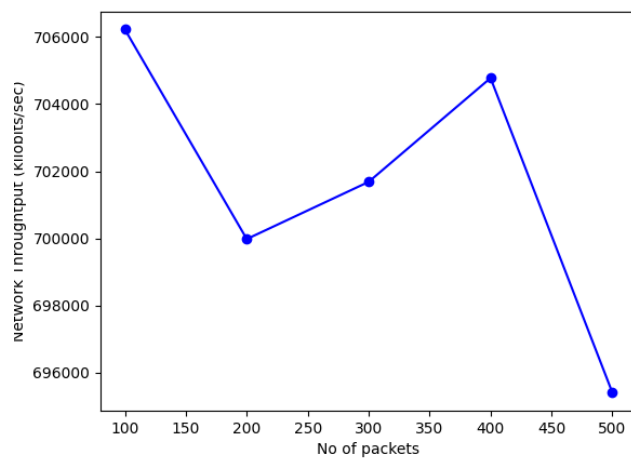
- Metrics Varying Number of Flows:

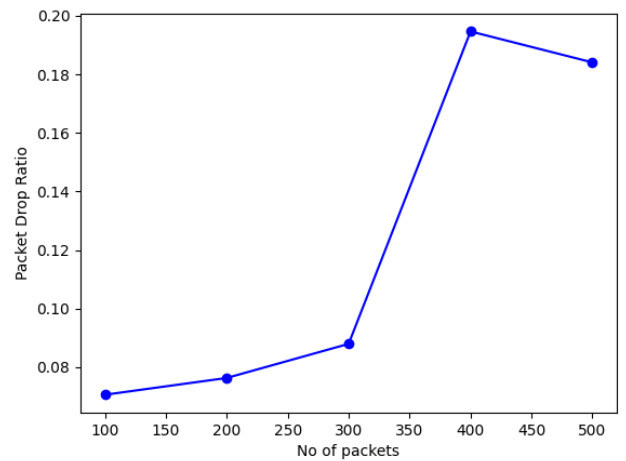
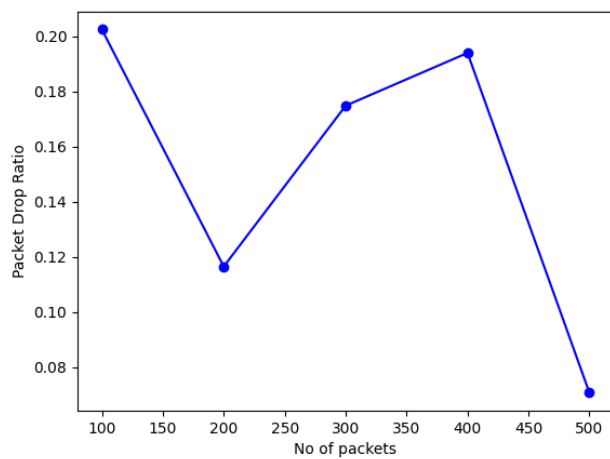
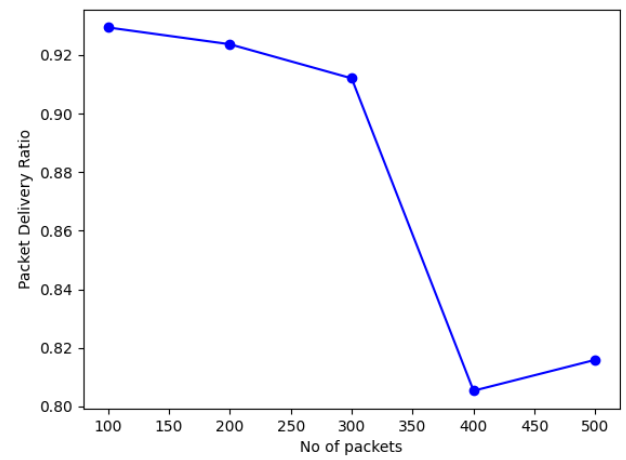
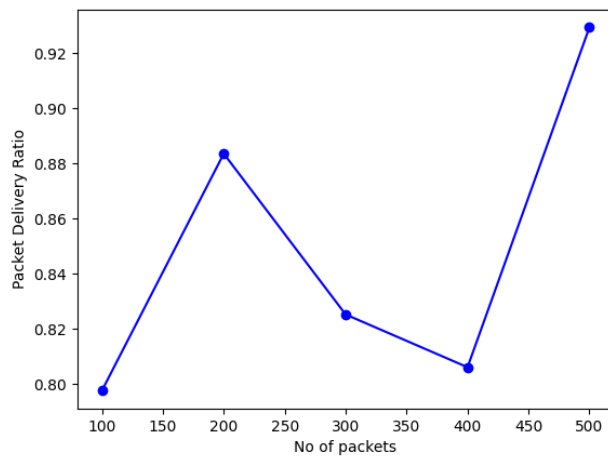
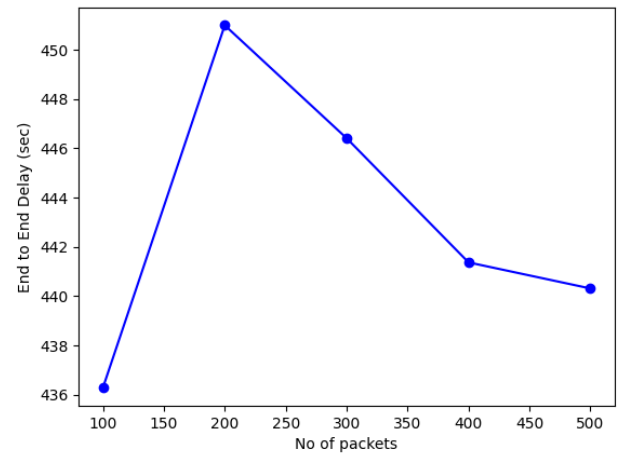
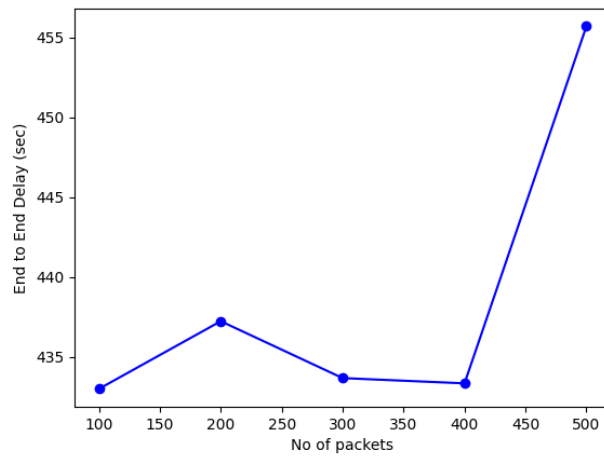


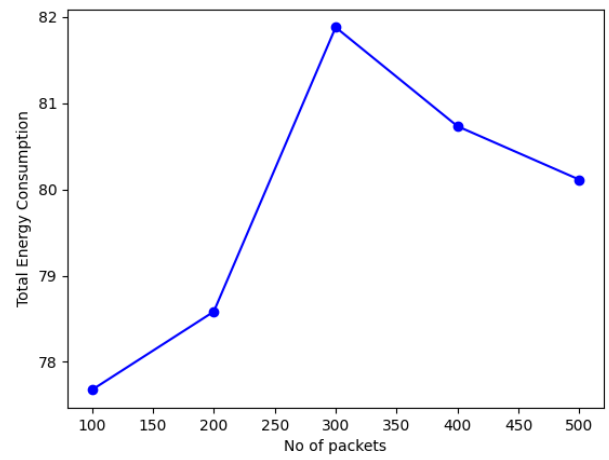
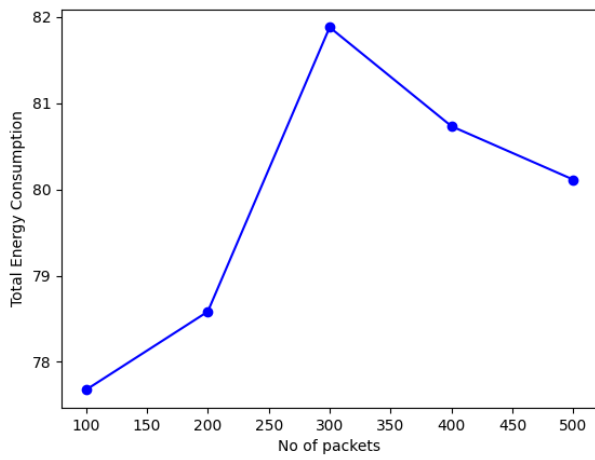




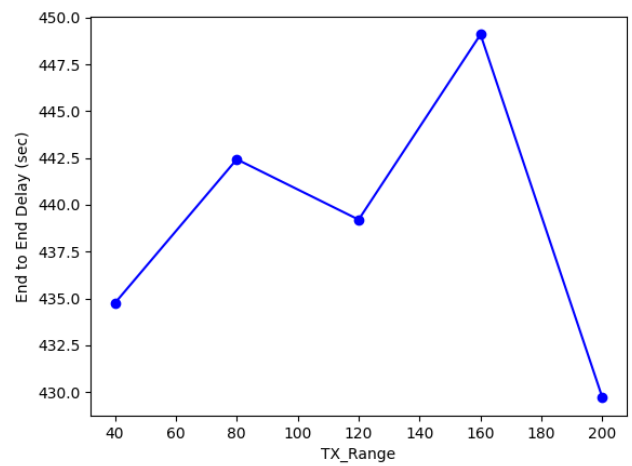
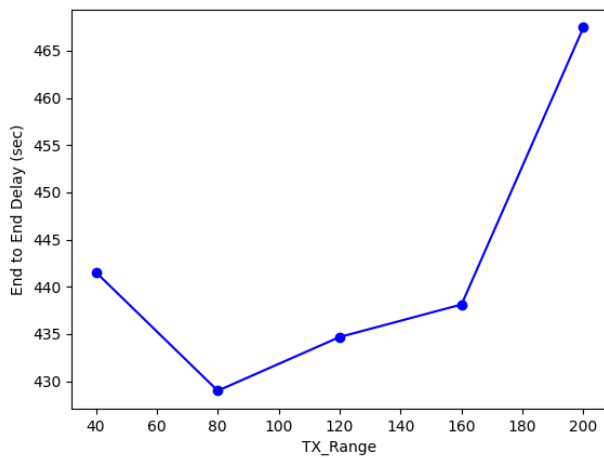
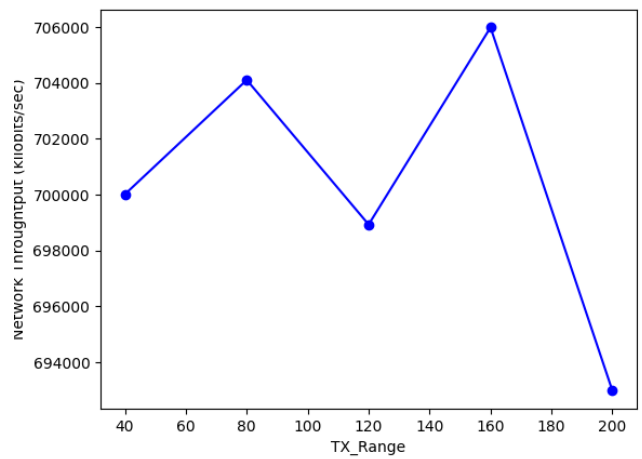
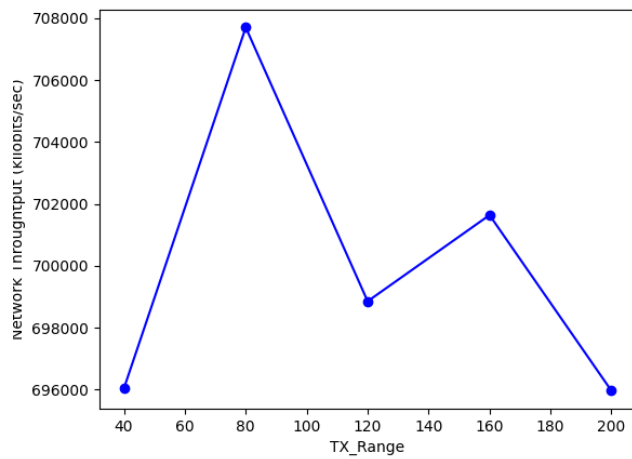
● Metrics Varying Number of Packet Per Second:

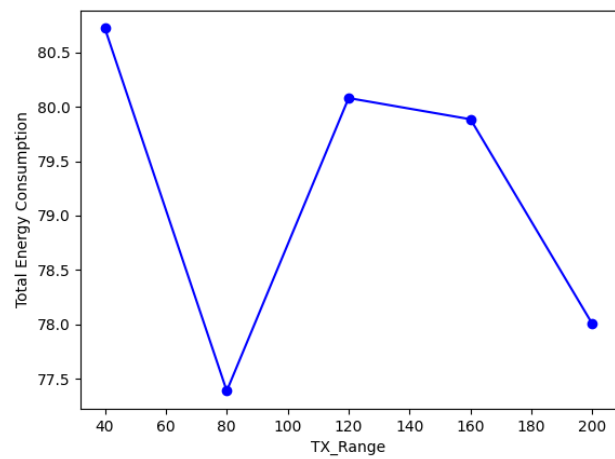
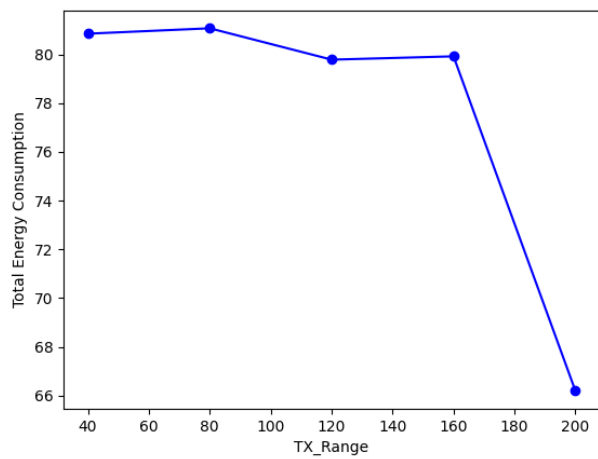
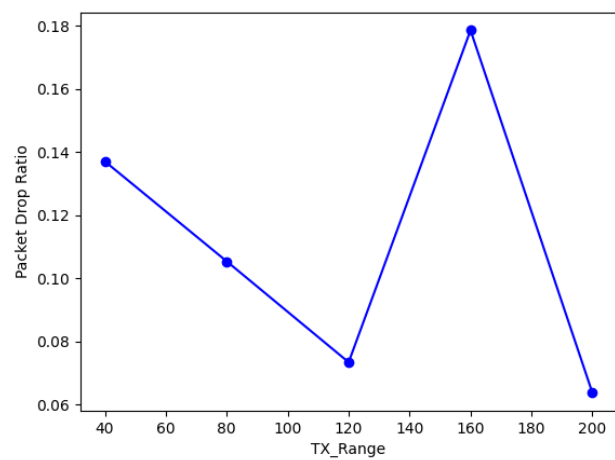
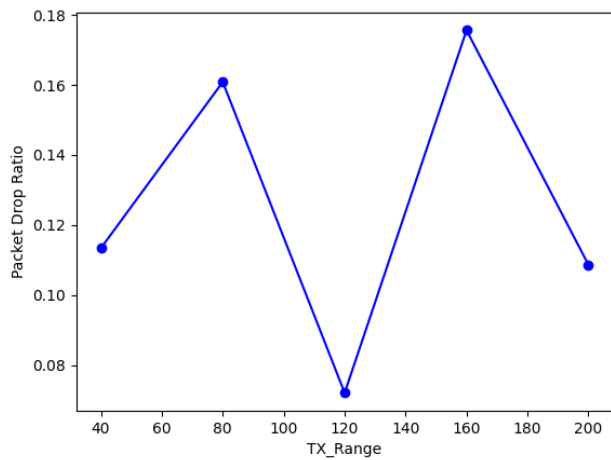
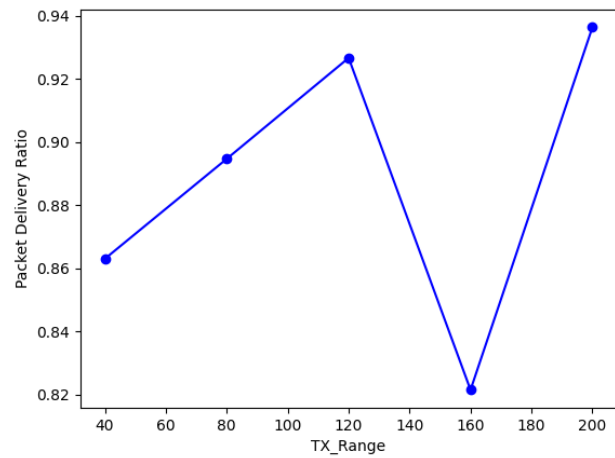
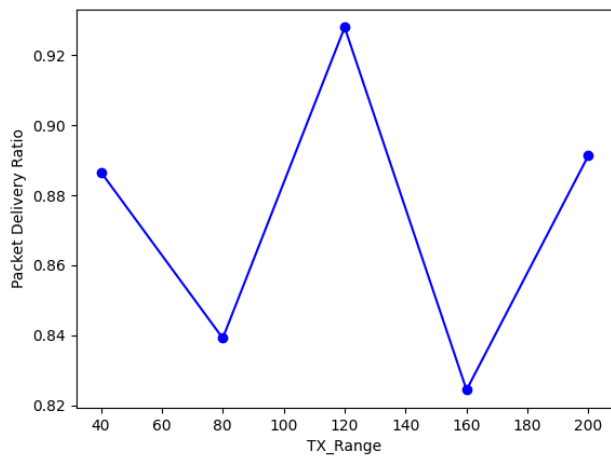






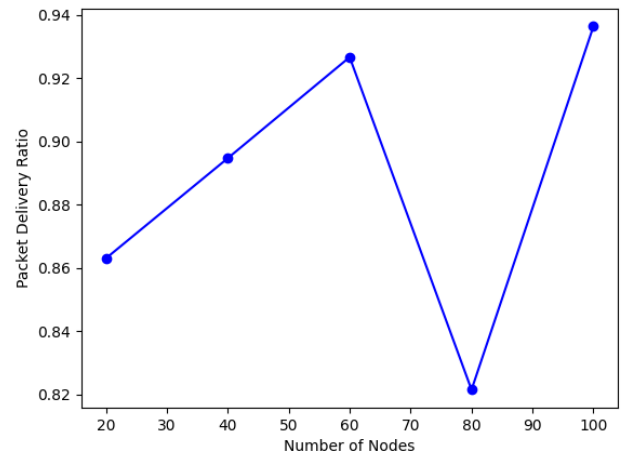
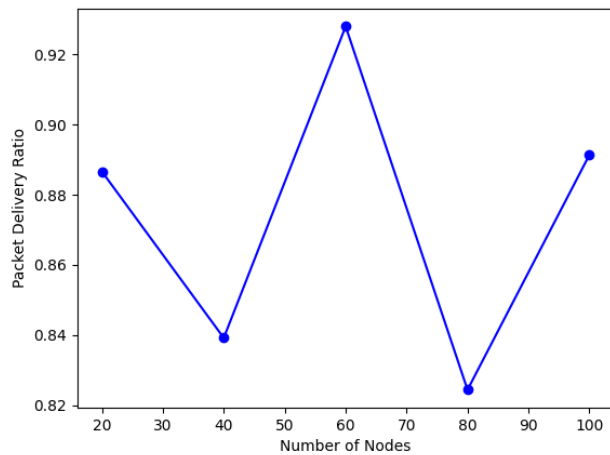
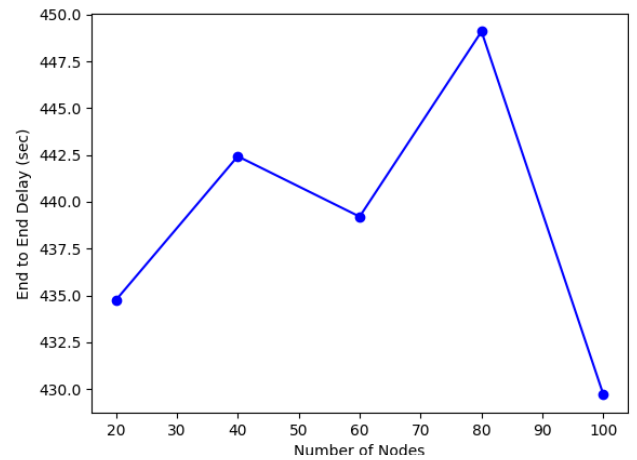
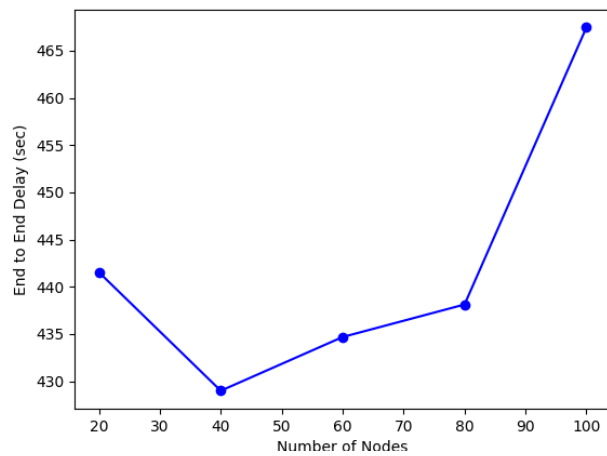
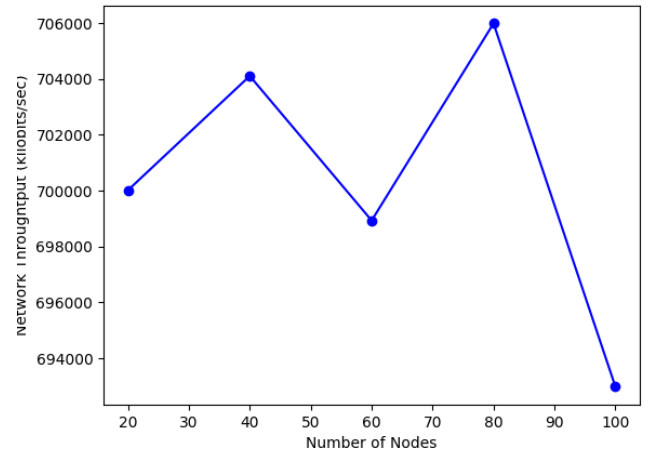
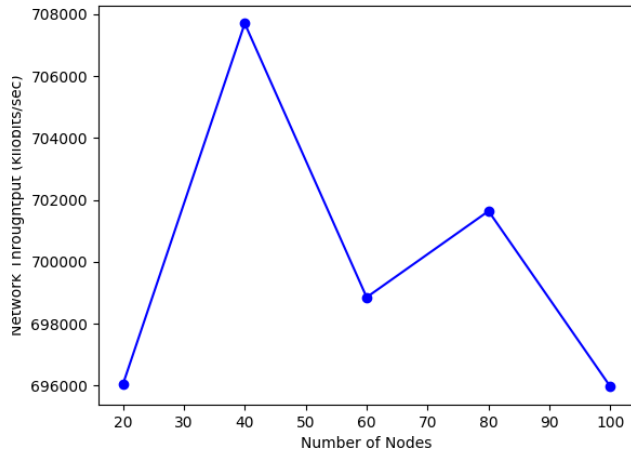
- Metrics Varying Coverage Area:

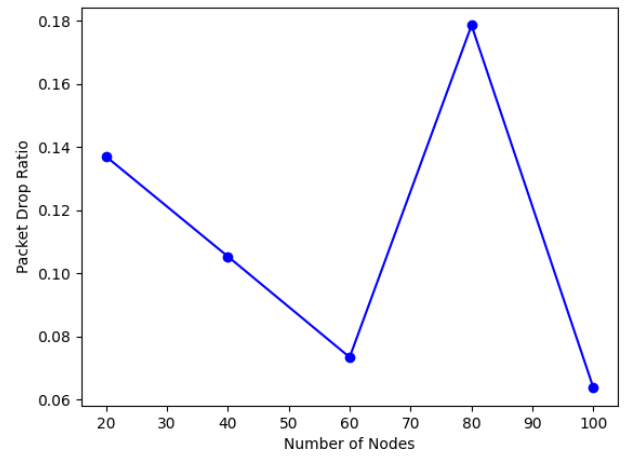
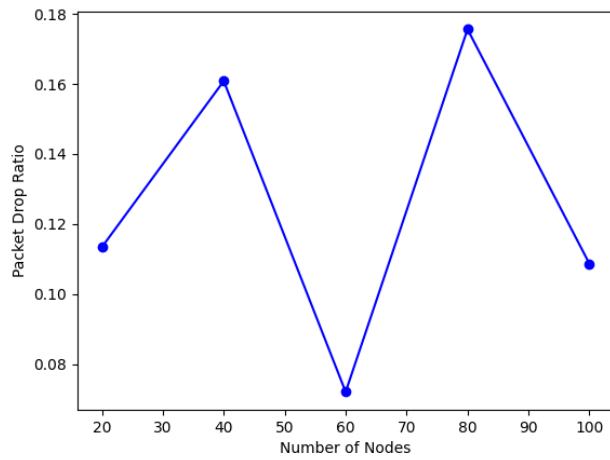




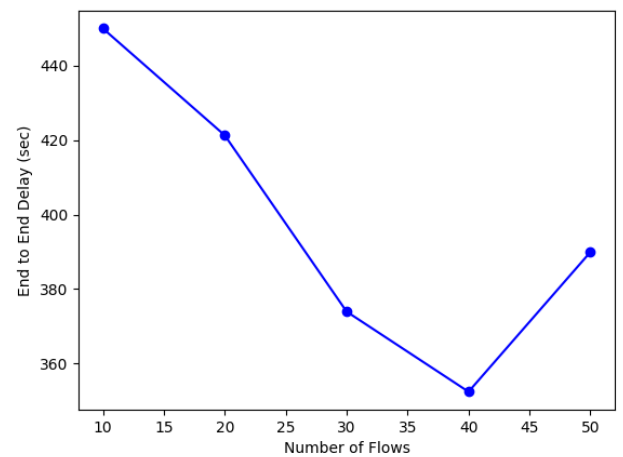
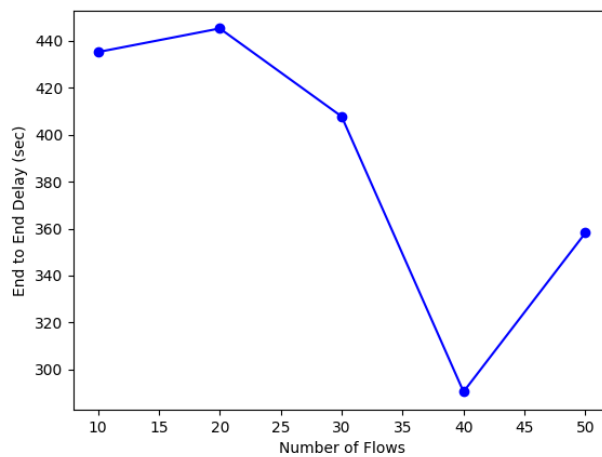
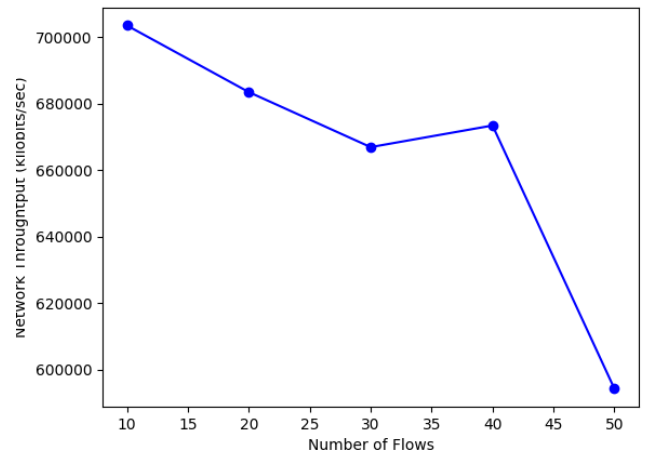
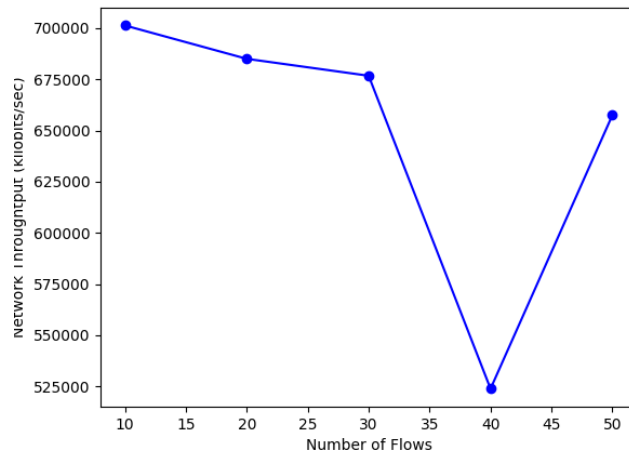
B. Wired (prev vs modified)

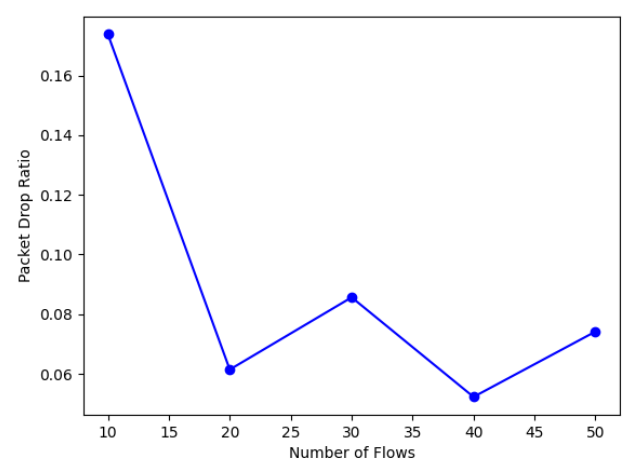
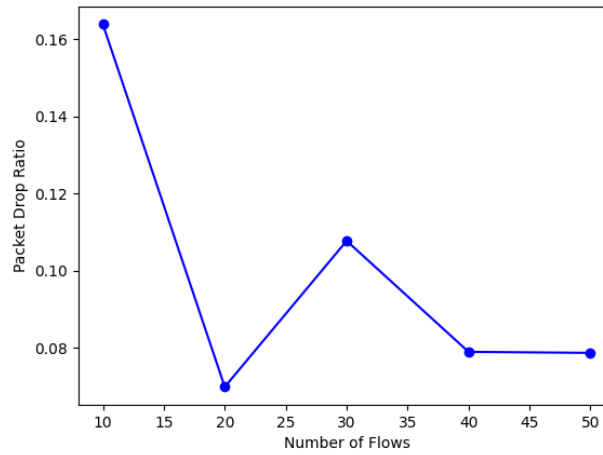
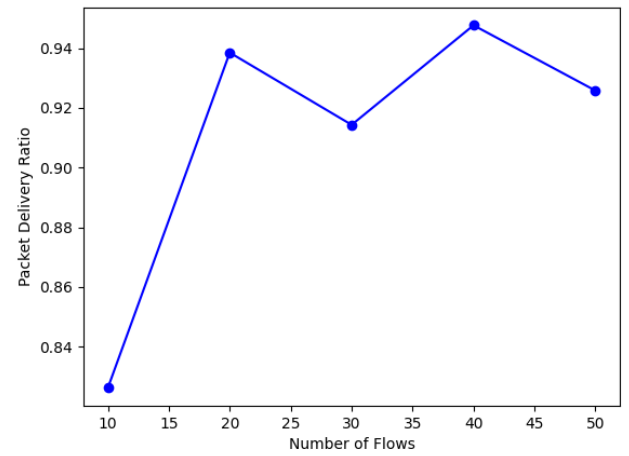
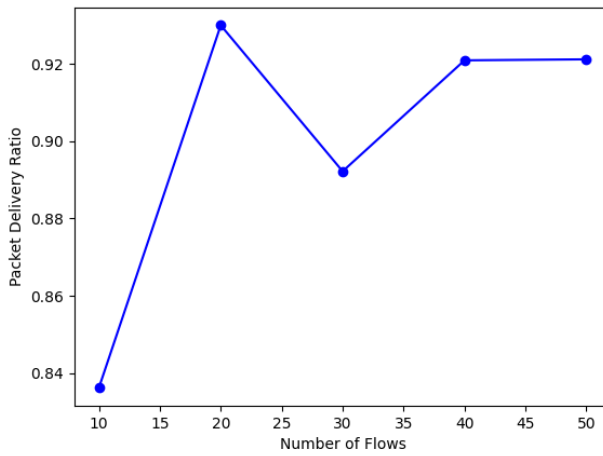
- Metrics Varying Number of Nodes:



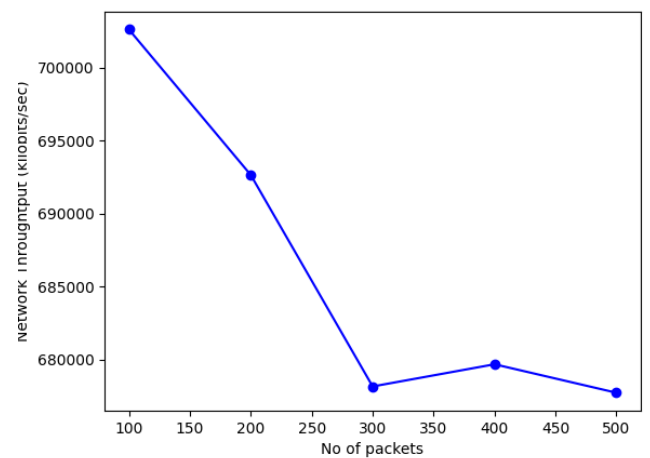
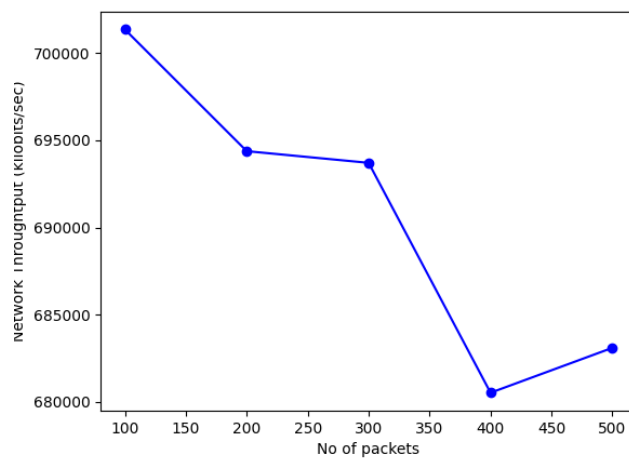


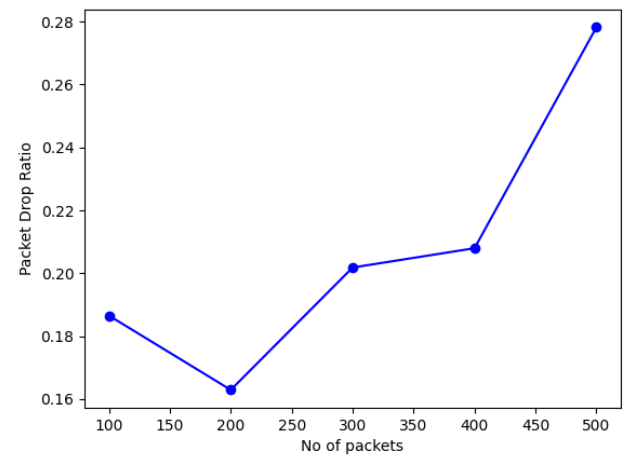
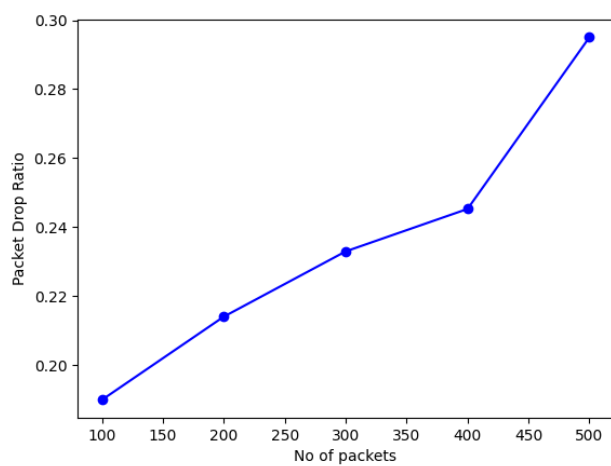
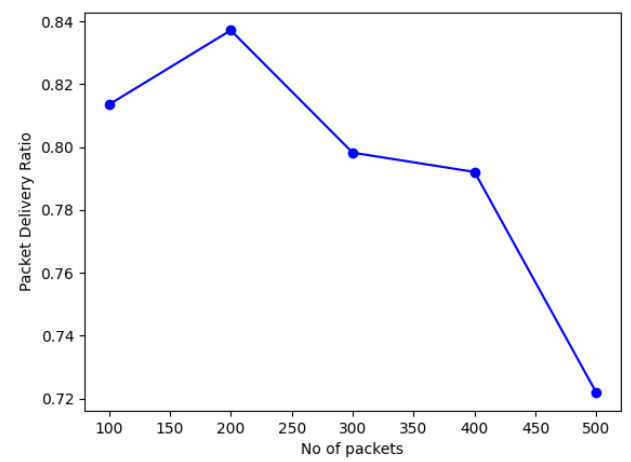
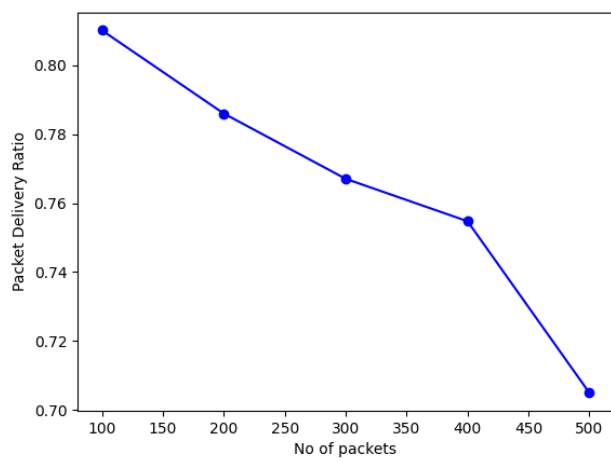
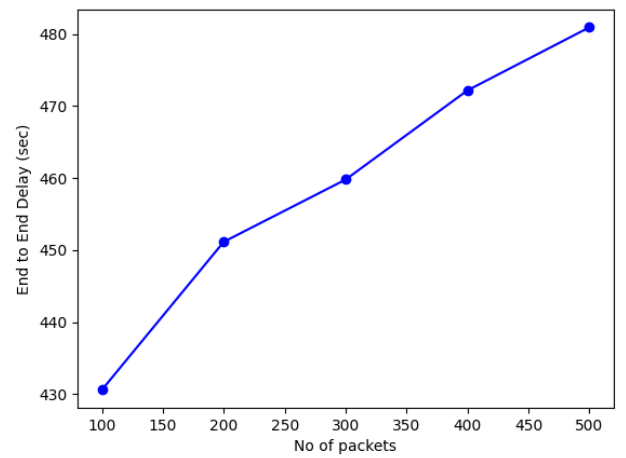
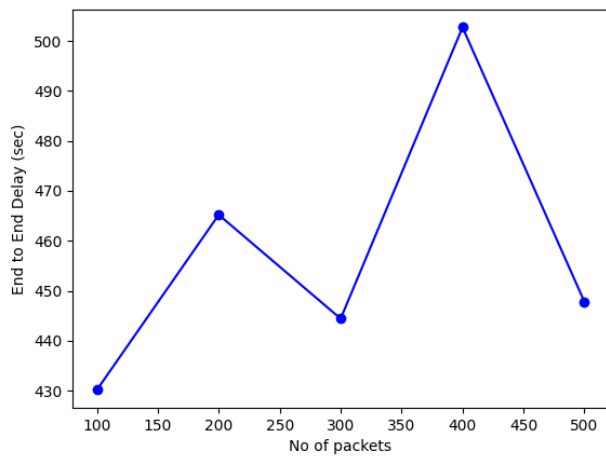
- Metrics Varying Number of Flows:





- Metrics varying Number of Packet Per Second:





5. Explanation

Wireless

1. Increase of nodes :

1. Throughput and end to end delay have decreased but the rate of decrease is less in modified version.
2. Packet delivery ratio has increased and packet drop ratio has decreased in modified version.
3. Total energy consumption didn't change at all.

B. Increase of flows :

1. Throughput has decreased in the modified version.
2. End to end delay has increased in the modified version.
3. The packet delivery ratio has decreased but the rate is lesser in the modified version. The packet drop has increased but lesser in the modified version.
4. Total energy consumption didn't change at all.

C. Increase of Packets :

1. Throughput has increased in the modified version.
2. End to end delay has decreased in the modified version.
3. The packet delivery ratio has decreased more than the original one. The packet drop has increased more than the original version.
4. Total energy consumption didn't change at all.

D. Increase of Coverage Area

1. Throughput and End to end delay has increased in the modified version.
2. The packet delivery ratio has increased more than the original one. The packet drop has decreased more than the original version.
3. Total energy consumption changed a little bit between them.

Wired

A. Increase of nodes :

1. Throughput changed a little bit in wired and end to end delay has increased but the rate of decrease is less in modified versions.
2. Packet delivery ratio has increased and packet drop ratio has decreased in modified versions.

B. Increase of flows :

1. Throughput and end to end delay has decreased but the rate of decrease is lesser in modified versions.
2. The packet delivery ratio has increased in the modified version. The packet drop has decreased in the modified version.

C. Increase of Packets :

1. Throughput has decreased in the modified version.
2. End to end delay has increased but the rate is less in the modified version.
3. The packet delivery ratio has increased more than the original one. The packet drop has decreased more than the original version.

Summery

Throughput and End to end delay has improved than the previous algorithm. Overall, packet delivery ratio has increased in the modified version. Packet drop ratio has decreased vice-versa. In the same energy consumption, we have attained better performance in the wireless system by cubic fit algorithm.